

☆ 게임프로그래밍

# HTML Game Example

## CANVAS 예제

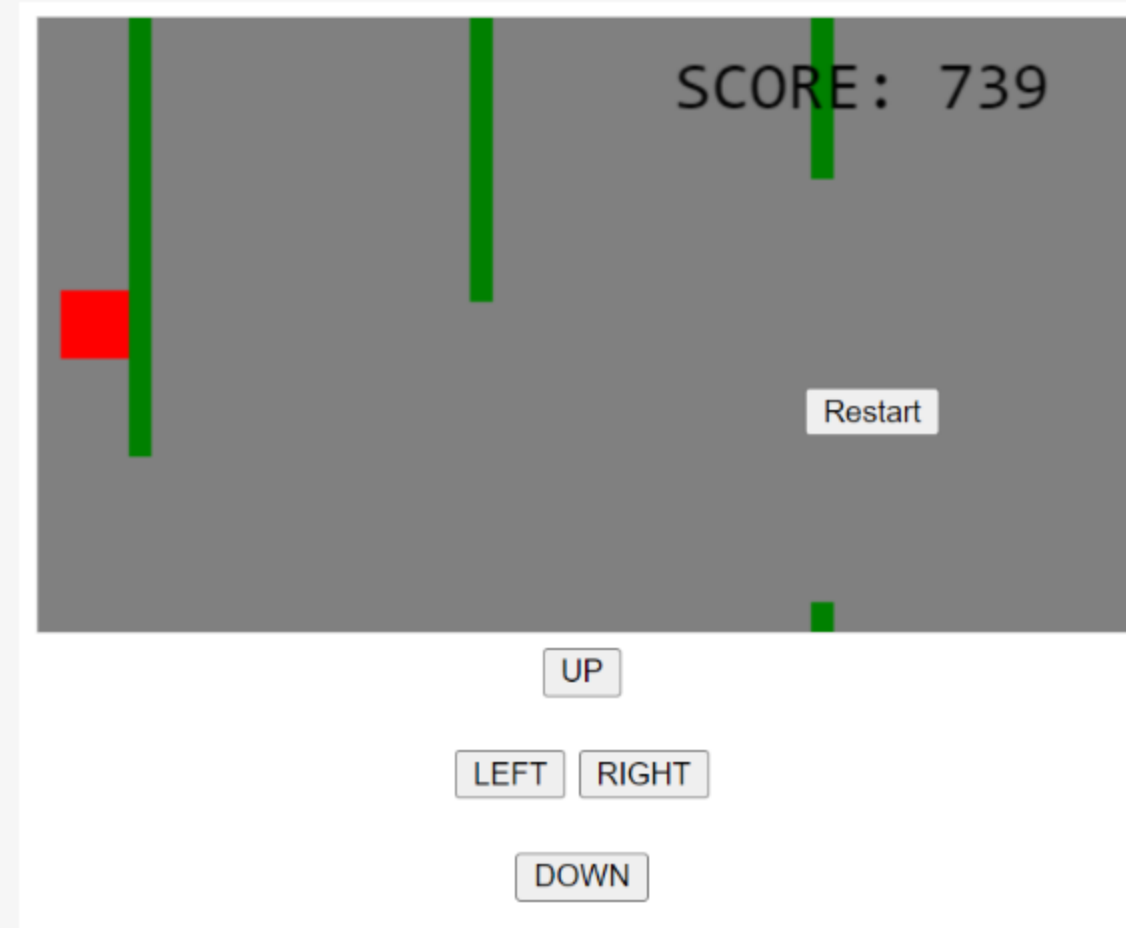
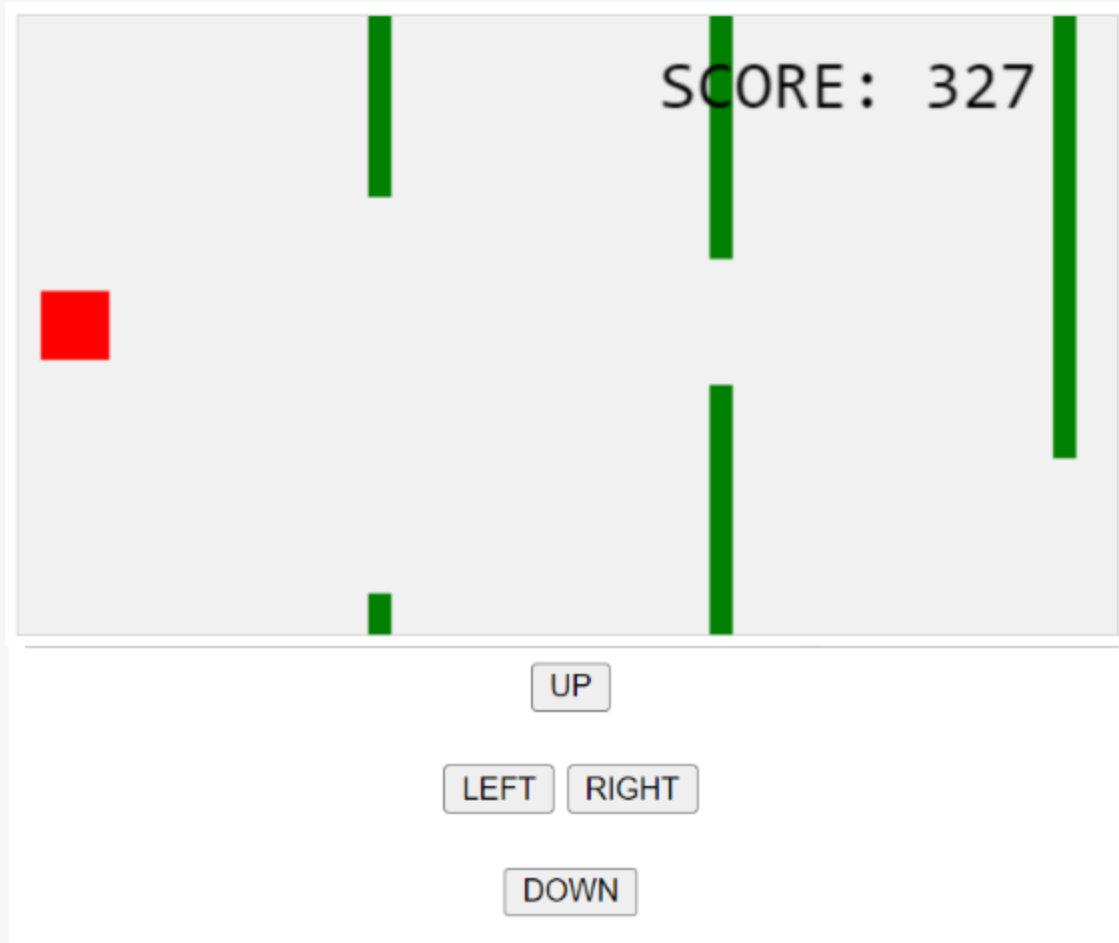
김정원(2021763022)

2023-11-08



## # 게임소개

마우스를 이용한 이동구현 :: 장애물 피하기 게임

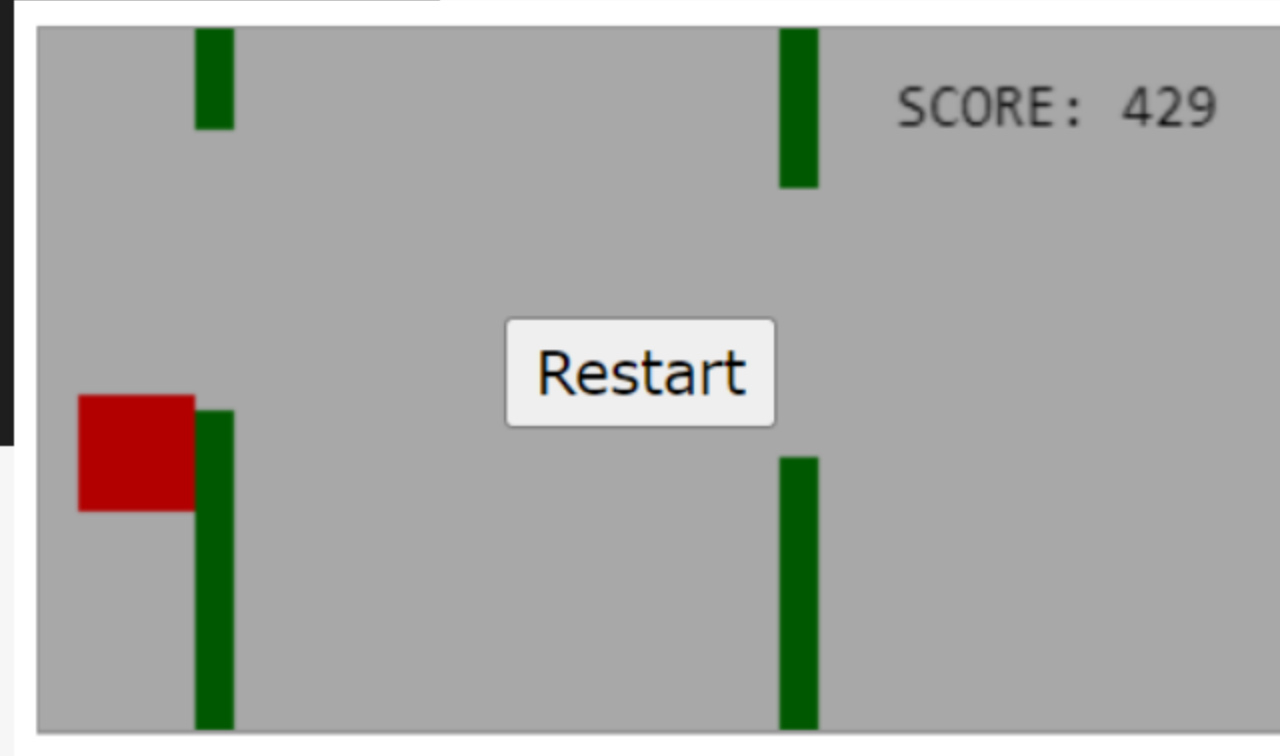


# 1. style

## 웹페이지의 CSS 스타일을 포함하는 HTML 태그

- 페이지의 모든 <canvas> 요소를 대상으로 하는 CSS 선택자입니다.

```
<style>
  canvas {
    border: 1px solid #d3d3d3;
    background-color: #f1f1f1;
  }
  #restart-button {
    position: absolute;
    display: none;
    top: 20%;
    left: 22%;
    transform: translate(-50%, -50%);
  }
</style>
```



## 2. startGame() :: onload (javascript)

### 문서의 모든 콘텐츠가 로드된 후 발생하는 이벤트

- 문서에 포함된 모든 콘텐츠가 로드된 후에 실행되기에 불필요한 로딩시간이 추가될 수 있음.
- 동일한 문서에 오직 onload는 하나만 존재해야 함.

```
<body onload="startGame()">
```

```
function startGame() {  
    myGamePiece = new component(30, 30, "red", 10, 120);  
    myScore = new component("30px", "Consolas", "black", 280, 40, "text");  
    myGameArea.start();  
}
```



SCORE: 382

```
function component(width, height, color, x, y, type) {  
    this.type = type;    this.width = width;    this.height = height;  
    this.speedX = 0;    this.speedY = 0;  
    this.x = x;    this.y = y;
```

### 3. myGameArea :: 객체 리터럴(object literal)

객체 내부에서 속성 이름과 값을 설정하는 방식을 나타냅니다.

- : (콜론): 속성 이름과 값 사이의 구분자입니다. 이것은 속성 이름과 해당 값 사이를 연결합니다.
- 객체 리터럴을 사용하면 여러 속성과 메서드를 하나의 객체로 그룹화하고 초기화할 수 있으며, 이를 통해 코드를 구조화하고 데이터와 동작을 객체 단위로 관리할 수 있습니다.

```
var myGameArea = {  
  canvas : document.createElement("canvas"),  
  start : function() {  
    this.canvas.width = 480;  
    this.canvas.height = 270;  
    this.context = this.canvas.getContext("2d");  
    document.body.insertBefore(this.canvas, document.body.childNodes[0]);  
    this.frameNo = 0;  
    this.interval = setInterval(updateGameArea, 20); },  
  clear : function() {  
    this.context.clearRect(0, 0, this.canvas.width, this.canvas.height); },  
  stop : function() {  
    clearInterval(this.interval); }  
}
```

- document.body.insertBefore(this.canvas, document.body.childNodes[0]);

<canvas> 요소를 HTML 문서의 <body> 요소 내에 삽입하고, 이 요소를 <body>의 첫 번째 자식 요소로 위치시킵니다.  
이것은 게임 화면이 웹 페이지의 맨 처음에 나타날 수 있도록 합니다.

## 4. component \_ 생성자(constructor)

```
function component(width, height, color, x, y, type) {  
  this.type = type;   this.width = width;   this.height = height;  
  this.speedX = 0;    this.speedY = 0;  
  this.x = x;         this.y = y;  
  
  this.update = function() {  
    ctx = myGameArea.context;  
    if (this.type == "text") {  
      ctx.font = this.width + " " + this.height;  
      ctx.fillStyle = color;  
      ctx.fillText(this.text, this.x, this.y);  
    } else {  
      ctx.fillStyle = color;  
      ctx.fillRect(this.x, this.y, this.width, this.height);  
    }  
  }  
  
  this.newPos = function() { this.x += this.speedX; this.y += this.speedY; }  
  this.crashWith = function(otherobj) {  
    var myleft = this.x;    var myright = this.x + (this.width);  
    var mytop = this.y;     var mybottom = this.y + (this.height);  
    var otherleft = otherobj.x;    var otherright = otherobj.x + (otherobj.width);  
    var othertop = otherobj.y;    var otherbottom = otherobj.y + (otherobj.height);  
  
    var crash = true;  
    if ((mybottom < othertop) || (mytop > otherbottom) || (myright < otherleft) || (myleft > otherright)) { crash = false; }  
    return crash;  
  }  
}
```

update : 업데이트 함수

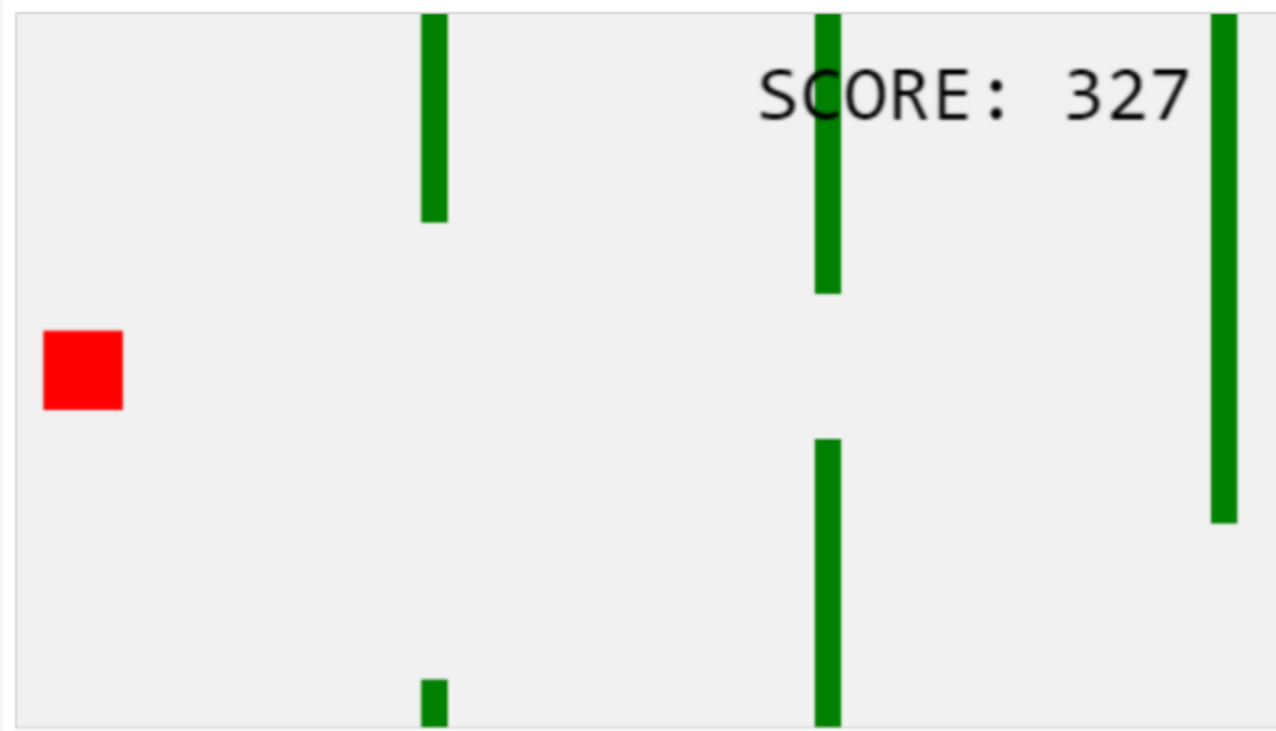
newPos : 게임 개체 위치 메서드

crashWith : 충돌 여부를 검사



## 5. updateGameArea()

게임 영역을 업데이트하는 주요 함수



물체 충돌 여부 확인 :: 게임을 중단하고 restart 버튼 생성  
프레임 관리 :: 프레임 수에 따라 초록색 장애물 생성  
업데이트 :: 장애물, 스코어, 플레이어의 위치 업데이트

```
function updateGameArea() {
    var x, height, gap, minHeight, maxHeight, minGap, maxGap;
    for (i = 0; i < myObstacles.length; i += 1) {
        if (myGamePiece.crashWith(myObstacles[i])) {
            myGameArea.stop();
            document.getElementById("restart-button").style.display = "block";
            endGame();
            return;
        }
    }
    myGameArea.clear();
    myGameArea.frameNo += 1;
    if (myGameArea.frameNo == 1 || everyinterval(150)) {
        x = myGameArea.canvas.width;
        minHeight = 20;    maxHeight = 200;
        height = Math.floor(Math.random() * (maxHeight - minHeight + 1) + minHeight);
        minGap = 50;    maxGap = 200;
        gap = Math.floor(Math.random() * (maxGap - minGap + 1) + minGap);
        myObstacles.push(new component(10, height, "green", x, 0));
        myObstacles.push(new component(10, x - height - gap, "green", x, height + gap));
    }
    for (i = 0; i < myObstacles.length; i += 1) {
        myObstacles[i].speedX = -1;
        myObstacles[i].newPos();
        myObstacles[i].update();
    }
    myScore.text = "SCORE: " + myGameArea.frameNo;
    myScore.update();
    myGamePiece.newPos();
    myGamePiece.update();
}

function everyinterval(n) {
    if ((myGameArea.frameNo / n) % 1 == 0) {return true;}
    return false;
}
```

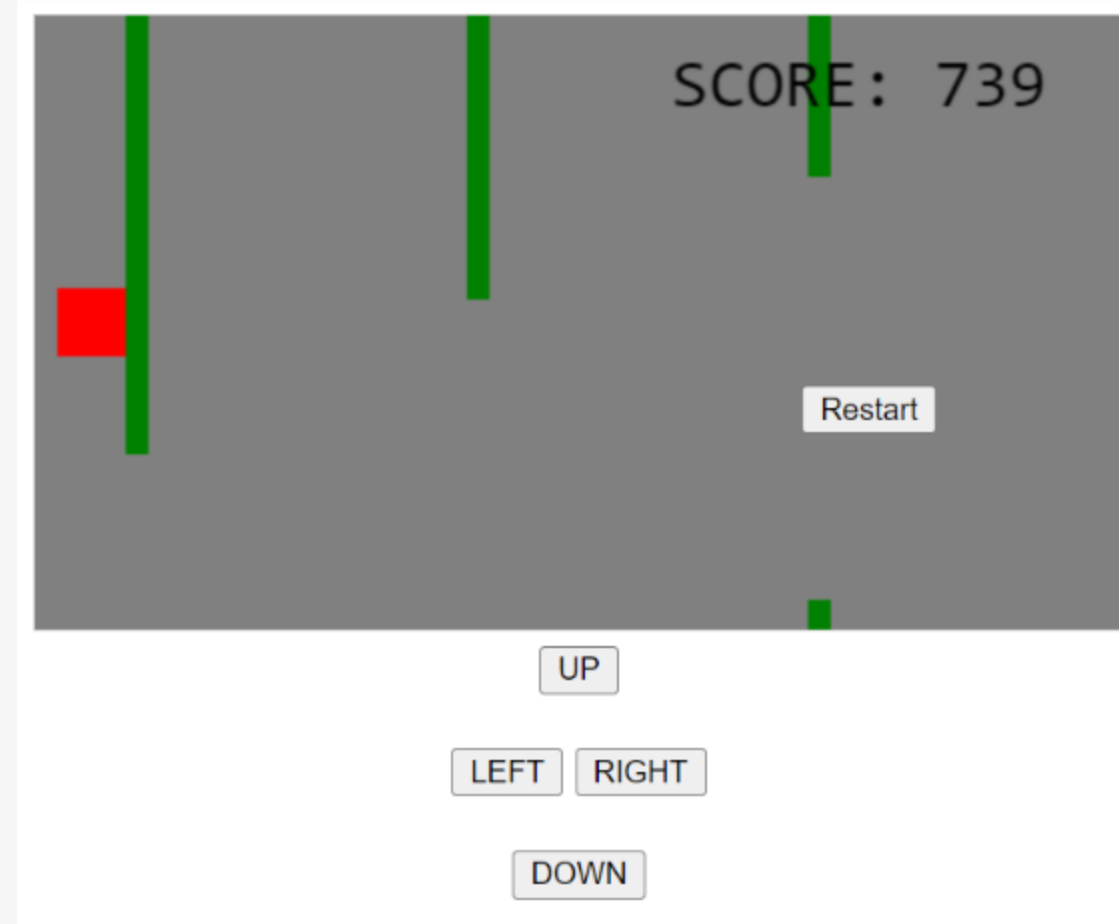
## 6. restartGame

### 게임 영역 초기화 및 재시작

```
function restartGame() {
  myGameArea.canvas.style.backgroundColor = "#f1f1f1";
  document.getElementById("restart-button").style.display = "none";
  myGameArea.clear();
  myObstacles = [];
  myGameArea.frameNo = 0;
  myGamePiece = new component(30, 30, "red", 10, 120);
  myGameArea.start();

  function endGame() {
    myGameArea.stop();
    myGameArea.canvas.style.backgroundColor = "gray";
    document.getElementById("restart-button").style.display = "block";
  }

  <div style="text-align:center;width:480px;">
    <button onmousedown="moveup()" onmouseup="clearmove()" ontouchstart="moveup()">UP</button><br><br>
    <button onmousedown="moveleft()" onmouseup="clearmove()" ontouchstart="moveleft()">LEFT</button>
    <button onmousedown="moveright()" onmouseup="clearmove()" ontouchstart="moveright()">RIGHT</button><br><br>
    <button onmousedown="movedown()" onmouseup="clearmove()" ontouchstart="movedown()">DOWN</button>
    <br><br> <button id="restart-button" onclick="restartGame()">Restart</button>
  </div>
```





## # 함수 실행 시점 정리

startGame()

MyGameArea.start()

updateGameArea() 20 밀리초마다 반복

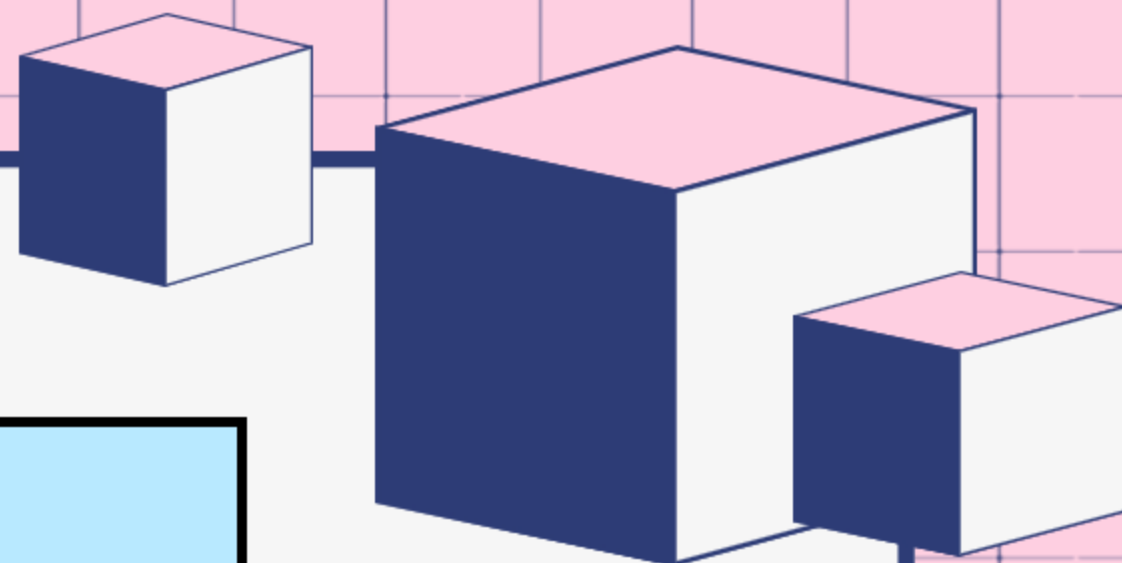
component.newPos()  
component.update()  
crashWith 충돌여부 확인

myGameArea.clear()

MyGameArea.stop()

endgame()

restartGame()



**THANK YOU.**